Modélisation de Ressources Termino-Ontologiques en OWL

Axel Reymonet, Jérôme Thomas, Nathalie Aussenac-Gilles

Modélisation de Ressources Termino-Ontologiques en OWL

Axel Reymonet[1,2], Jérôme Thomas[2], Nathalie Aussenac-Gilles[1]

[1] Équipe Conception de Systèmes Coopératifs, IRIT,
Université Paul Sabatier, 118 Route de Narbonne, F-31062 TOULOUSE CEDEX 9
reymonet@irit.fr, aussenac@irit.fr

[2] Division Technologie ACTIA,
25 Chemin de Pouvourville, BP 4215, 31432 TOULOUSE CEDEX 4
axel.reymonet@actia.fr, jerome.thomas@actia.fr

Résumé : Dans le cadre de recherches menées sur l'indexation sémantique, nous avons été conduits à nous interroger sur l'efficacité générale des modèles actuels de représentation des terminologies au sein des ontologies. Après avoir évoqué leurs limites actuelles, nous proposons un nouveau modèle pour manipuler une ressource termino-ontologique en OWL et nous en décrivons son implémentation prochaine au sein de l'environnement Protégé.
Mots-clés : Ontologie, Terminologie, Modèles de représentation, Indexation sémantique, Protégé

1 Introduction

Depuis le lancement du World Wide Web au début des années 90, les sites Web ont connu un essor fulgurant tant dans leur nombre (d'une trentaine de sites en 1992 à plus de cent millions fin 2006) que dans la diversité et la richesse de leur contenu. Le Web Sémantique est issu de la volonté de permettre une interprétation automatique du contenu de ces sites. Une solution envisagée est de manipuler une version enrichie du Web, complétée de métadonnées et d'annotations sémantiques permettant au système de produire des raisonnements ou d'appliquer des algorithmes. C'est dans ce contexte que nous avons étudié les moyens de représenter conjointement les concepts d'un domaine et les phénomènes linguistiques qui permettent de les repérer dans un texte. Après avoir présenté le contexte de recherche qui a suscité cette réflexion, nous évoquons les limites des modèles actuels de ressources termino-ontologiques (RTO). Nous proposons alors un nouveau modèle basé sur le standard W3C OWL. Pour finir, nous ferons une description sommaire de son implémentation prochaine.

Cette réflexion est conduite dans le cadre du laboratoire commun Autodiag, qui a reçu le soutien du Ministère de l'Industrie, de la Région Midi-Pyrénées et du programme européen FEDER. Les auteurs souhaitent remercier ces différents organismes pour leur aide.

2 Cadre de recherche

La problématique de cet article fait partie d'une réflexion en cours sur la recherche d'informations (RI) dans un ensemble de fiches de réparation automobile (environ 5000 synthèses d'experts en diagnostic automobile à partir d'analyses de cas réels rencontrés en garage). De façon plus précise, notre objectif consiste à optimiser les recherches dans cette base d'expériences : à partir d'une description des symptômes saisie manuellement en langue naturelle, nous cherchons à élaborer un outil capable de retrouver automatiquement toutes les fiches traitant d'un problème similaire sur un même modèle de véhicule [1]. C'est dans ce cadre que nous nous sommes intéressés à la notion d'indexation sémantique.

2.1 L'indexation sémantique

L'indexation sémantique peut être vue comme un cas particulier du processus d'indexation en recherche d'informations (RI). L'ingénierie documentaire en donne la définition suivante (AFNOR, 1987) : « L'indexation est un processus destiné à représenter par les éléments d'un langage documentaire ou naturel des données résultant de l'analyse du contenu d'un document ou d'une question. » Dans son acception informatique, elle consiste à parcourir les documents constituant la base de recherche, repérer toutes les informations associées et stocker le résultat afin de pouvoir retrouver celles-ci facilement (Prié, 2000). Aujourd'hui encore, la plupart des moteurs classiques de RI met en oeuvre des techniques statistiques appliquées au contenu lexical des textes, auxquelles viennent s'ajouter des critères basés sur le nombre et la popularité des références (Mothe, 2000). Cette solution a l'avantage d'être facilement automatisable, toutefois elle ne permet pas de représenter les informations à un niveau sémantique. Dans le cas d'une ressource aussi ouverte que l'Internet, les moteurs de recherche pêchent surtout par leur manque de précision, directement lié à leur incapacité à lever les ambiguïtés sémantiques des requêtes (Berners-Lee, 1999).

Pour pallier ces problèmes, le principe de l'indexation sémantique consiste à se placer à un niveau conceptuel pour le document. Les objets manipulés ne se réduisent plus à des chaînes de caractères pondérées mais un lien est fait entre celles-ci et les notions qu'elles désignent. L'index sert alors à stocker la présence de ces notions dans chacun des textes de la base de recherche. Ce passage à un niveau d'abstraction supérieur s'accompagne de certaines contraintes : disposer d'un modèle des connaissances présentes dans l'ensemble des documents et connaître les expressions linguistiques qui leur sont associées. Il est donc nécessaire, pour une indexation sémantique, de manipuler les notions d'ontologie et de terminologie, développées par la suite.

2.2 Ontologie et terminologie, convergence et complémentarité

La notion d'ontologie a été redéfinie au gré des débats dont elle a fait l'objet. Nous retiendrons la définition suivante (Charlet, 2002) : « Une ontologie est une spécification normalisée représentant les classes des objets reconnus comme existant dans un

[1] Pour plus de détails, voir (Reymonet *et al.*, 2006).

domaine. » L'ontologie doit être exploitable par un ordinateur tout en faisant sens pour les humains, définir de manière normalisée et consensuelle les concepts d'un domaine tout en répondant aux besoins de l'application visée.

Le rôle des ontologies dans un cadre de recherche d'informations se situe à la charnière entre les connaissances et leur expression dans la langue : elles doivent à la fois être un support à la formulation de recherches et une ressource pour définir des métadonnées destinées à annoter des documents. Plusieurs travaux comme (Maedche, 2002) mentionnent d'ailleurs la nécessité d'associer un lexique indépendant à une ontologie, de manière à étiqueter concepts et relations, le tout formant une ontologie à composante lexicale.

Au sens classique, une terminologie liste les termes d'un domaine, et pour chacun d'eux, propose une fiche qui en décrit les usages, la (ou les) signification(s), ainsi que les relations entretenues avec des termes sémantiquement et/ou syntaxiquement proches. Avec la mise sur support informatique et la diversification des usages des terminologies dans les années 90, les terminologues se sont interrogés sur les notions de termes et de concepts, et sur leur articulation.

Les bases de connaissances terminologiques (BCT) sont un des fruits de cette réflexion (Meyer *et al.*, 1992). Elles constituent un enrichissement significatif car leur modèle original différencie un niveau linguistique d'un niveau conceptuel : on accède par les termes du domaine à une modélisation conceptuelle qui donne sens à ces termes (Aussenac-Gilles & Condamines, 2001). Leur structure est proche de celle d'une ontologie (Szulman *et al.*, 2002) mais leur contenu n'a pas d'ambition ontologique. Au contraire, un tel modèle rend compte des concepts tels qu'ils se dégagent de l'étude de la langue, en restituant une sémantique qui reste au plus près de l'usage linguistique.

Pratique terminologique, mise au point de BCT et modélisation d'ontologie en lien avec la langue remettent en question certaines hypothèses structuralistes des années 40 comme la non ambiguïté des termes dans chaque domaine, ou leur stabilité d'usage et de sens. Elles soulignent aussi la diversité des outils permettant le repérage de concepts dans la langue : patrons d'extractions, contexte grammatical, contexte sémantique … La prise en compte du lexique est sans nul doute un moyen insuffisant pour accéder au concept mais n'en constitue pas moins la base de toutes les approches classiques.

2.3 Normes et standards de représentation

Dans un souci d'interopérabilité des ressources terminologiques d'une part et ontologiques d'autre part, nous avons souhaité respecter autant que possible les formats les plus conseillés, que nous présentons ci-après.

2.3.1 La norme TMF pour la description de terminologies

La norme ISO 16642 définit l'environnement TMF (Terminological Markup Framework) qui permet de décrire tous les éléments d'une terminologie avec un langage formel (Romary, 2001). Celui-ci est constitué d'un méta-modèle et d'un ensemble de contraintes sur les catégories de données utilisées pour représenter les propriétés de chaque terme. Le respect d'un tel format a l'avantage de garantir la compatibilité mutuelle de deux TML (Terminological Markup Language) de syntaxe différente. Le méta-

modèle de TMF représente la structure sous-jacente d'une terminologie sur plusieurs niveaux :

– les informations sémantiques (le concept)
– les réalisations linguistiques (les langues dans lesquelles est exprimé le concept)
– les informations lexicales (les termes associés au concept dans une certaine langue)

Pour décrire un terme, TMF recommande le recours aux catégories de données définies par la norme ISO 12620. Parmi les différentes sortes d'information, on peut trouver le type du terme, les informations grammaticales (catégorie syntaxique, genre, nombre …), les usages, la formation (provenance, étymologie), la prononciation ou la morphologie.

Pendant la phase de conception de la terminologie, les objectifs applicatifs influencent directement le processus de sélection des propriétés de terme utiles. Il faut néanmoins prendre en compte l'équilibre souhaité entre le niveau d'expressivité de la terminologie et la complexité des traitements ultérieurs [2]. Dans le cadre de notre étude, nous choisissons de nous restreindre à la partie lexicale et textuelle d'un terme. Nous représenterons donc principalement le terme à travers ses usages (textes dans lesquels il apparaît et position exacte de ses occurrences).

Il est bon de noter ici que même dans un cas de monosémie du domaine, la position d'un terme dans un texte est forcément reliée directement à celui-ci et non au concept qu'il désigne. Dans un contexte d'indexation sémantique, on pourrait penser que seule la localisation du concept nous importe, et pas celle des termes associés. Toutefois, nous considérons que l'ontologie est une représentation qui évolue avec le temps : l'apparition de nouveaux textes à indexer peut entraîner l'ajout ou la modification de concepts, ce qui aura pour conséquence probable la réorganisation des relations entre termes et concepts. Un terme désignant à l'origine un concept C sera par exemple associé à un nouveau concept C' ; il est alors nécessaire de pouvoir visualiser le contexte d'utilisation de ce terme (et non celui de tous les termes associés à C) .

2.3.2 Le standard OWL pour les ontologies

Dans le cadre du Web Sémantique, il est capital d'avoir un formalisme commun pour la représentation d'ontologies, afin de permettre une meilleure interopérabilité dans le partage, la modification et l'intégration de telles structures. A cet égard, RDFS et OWL sont considérés comme les langages les plus adéquats car ils sont issus de recommandations du W3C et bénéficient d'une expressivité adaptée aux besoins de chacun [3]. Tous deux s'appuient sur le langage de balisage XML, élément fondamental du Web Sémantique.

OWL est une évolution du langage Web DAML+OIL qui s'appuie sur RDFS. Il a été conçu « pour représenter explicitement la signification des termes des vocabulaires [au sens de la logique des prédicats] et les relations entre ces termes » (W3C, 2004). OWL dépasse RDFS par ses capacités à représenter une ontologie qui soit interprétable automatiquement. En effet, OWL introduit la possibilité pour une machine de raisonner sur

[2] Généralement, plus l'information stockée sera riche, moins les traitements devront être lourds.

[3] On aurait pu citer également XTM (XML Topic Maps) mais (Baget *et al.*, 2004) ne le considère pas comme un langage de définition d'ontologies mais comme un langage d'annotation de ressources.

la base de connaissances, ce qui lui permet d'inférer des connaissances implicites et de détecter d'éventuelles incohérences. De plus, le vocabulaire d'OWL s'avère plus riche que celui de RDFS car il rajoute des relations entre classes, des propriétés de cardinalité, d'égalité, la définition de classe par énumération. OWL permet de gérer des niveaux de complexité différents à travers trois sous-langages à l'expressivité croissante :

- OWL Light, sous-ensemble minimal destiné à la construction de taxinomies,
- OWL DL (Description Logics), à la fois beaucoup plus expressif qu'OWL Light et garant de la complétude et de la décidabilité des calculs,
- OWL Full avec la liberté syntaxique de RDFS mais sans la complétude des calculs.

Afin de bénéficier de toute la richesse sémantique d'OWL et d'anticiper l'utilisation ultérieure de raisonneurs, nous avons choisi de nous intéresser plus particulièrement aux différentes façons de modéliser la partie terminologique d'une RTO en OWL-DL.

3 Modèles de représentation des termes en OWL

Dans cette section, nous décrivons et analysons quelques modèles permettant de représenter conjointement les parties ontologique et terminologique d'une RTO dans le standard OWL. Nous verrons notamment en quoi ceux-ci pourraient être améliorés, ce qui nous amènera dans la section suivante à une proposition originale de modèle.

3.1 La modélisation du terme comme attribut du concept

En OWL, les éléments de base d'une ontologie sont matérialisés de la façon suivante :

- les concepts de l'ontologie sous forme de `owl:Class`,
- les attributs de concepts sous forme de `owl:DatatypeProperty`,
- les relations entre concepts sous forme de `owl:ObjectProperty`.

OWL a été élaboré dans l'idée de servir à l'indexation de ressources sur le Web. Il permet donc de représenter le lexique sous la forme duquel un concept pourra apparaître dans un document ou dans l'interaction avec l'utilisateur. Pour modéliser les termes désignant ce concept, OWL associe à la classe correspondante une (ou plusieurs) chaîne(s) de caractères au moyen d'une propriété d'annotation, `rdfs:label`.

Plusieurs problèmes découlent de ce choix de modélisation. Tout d'abord, un terme ainsi représenté n'a pas d'existence en tant que tel. Par conséquent, on ne peut pas lui associer directement de propriétés qui lui sont pourtant intimement liées (comme la position de ses occurrences dans un texte), il faut passer par le concept qu'il désigne. Il est alors impossible de respecter le méta-modèle TMF qui prône -entre autres- la dissociation des informations conceptuelles et lexicales. Enfin, une indexation sémantique se basant sur un tel modèle cherche à retrouver la trace de classes d'objets dans les textes, et non celle d'objets. Implicitement, cette indexation assimile chaque terme identifié à une nouvelle instance du concept associé. En effet, la représentation des instances en OWL ne permet pas d'associer à l'instance l'occurrence d'un terme particulier dans un document. Or cela pourrait être utile pour comparer, assimiler ou distinguer des occurrences, par exemple dans le cas d'anaphores.

3.2 L'emploi de propriétés d'annotations structurées : Terminae

(Szulman *et al.*, 2002) décrit à travers l'utilisation du logiciel Terminae, une méthode de construction d'ontologies orientée vers les textes. Pour assurer une traçabilité des concepts vers les textes qui les évoquent, Terminae permet de gérer une terminologie du domaine, chaque terme donnant lieu à une fiche terminologique.

Dans un souci de standardisation des exports, (Szulman & Biébow, 2004) aborde le problème des insuffisances d'OWL à représenter de façon détaillée les termes et propose d'enrichir le modèle (sans extension de la syntaxe OWL) afin que des paramètres des termes, comme leurs synonymes et leurs occurrences dans les textes, puissent être pris en compte. A cet effet, les auteurs utilisent la structure de propriété d'annotation d'OWL (`owl:AnnotationProperty`) qui permet d'ajouter des informations particulières à une classe. En figure 1, on peut voir comment la représentation conjointe d'un concept et d'un terme se traduit alors en OWL.

```
<owl:AnnotationProperty rdf:about="&terminae;term" />
<owl:AnnotationProperty rdf:about="&terminae;synonym" />
<owl:AnnotationProperty rdf:about="&terminae;occurrence" />
[...]
<owl:Class rdf:ID="code_défaut">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#observation_anormale" />
    </rdfs:subClassOf>
    <terminae:term>code défaut</terminae:term>
    <terminae:synonym>CD</terminae:synonym>
    <terminae:occurrence> Ident NB1H000C-divers- 1 : occ n°1
                          Code défaut : 188 </terminae:occurrence>
    <terminae:occurrence> Ident NB1H000F-divers- 2 : occ n°2
                          Le CD 498 apparait</terminae:occurrence>
</owl:Class>
```

FIG. 1 – *Export OWL de Terminae*

Cette structuration permet de stocker un plus grand nombre d'informations sur chacun des termes que l'utilisation de la seule propriété `rdfs:label`. Nous avons d'ailleurs retenu dans un premier temps cette solution pour la construction et le stockage d'une RTO du diagnostic automobile. Toutefois, ce choix de modélisation qui consiste à rajouter autant de propriétés d'annotation au concept que d'attributs de terme rend impossible la manipulation du terme indépendamment du concept associé : on ne peut par exemple accéder aux occurrences d'un terme particulier. En outre, à la différence des propriétés d'objet et des propriétés de type de données, les propriétés d'annotation ne sont pas prises en compte par un raisonneur OWL. Il sera donc impossible de conduire des inférences logiques sur les liens associant termes et concepts, sauf à modifier le comportement de ces raisonneurs.

3.3 L'assimilation du terme à une instance de concept

Une autre approche, commune en extraction d'information, consiste à considérer le terme comme instance du concept auquel il est associé. GATE, plate-forme incontournable du domaine [4], adopte ce choix de modélisation (Bontcheva *et al.*, 2004). Même s'il ne manipule pas une ontologie directement en OWL, GATE dispose néanmoins d'une fonctionnalité d'import / export sous ce format. Il permet notamment de reconnaître dans les textes des listes d'instances (appelées *gazetteers*), de définir puis de projeter des patrons d'extraction grâce auxquels des expressions linguistiques sont associées à des instances de concepts (cette étape est réalisée par l'application de règles JAPE). C'est ainsi que les expressions extraites accèdent au statut de terme.

Ce choix de considérer le terme comme une instance de concept n'est pas sans poser certains problèmes théoriques et pratiques. En effet, si l'on revient aux définitions de base de l'ingénierie des connaissances, un concept est une représentation mentale destinée à regrouper un ensemble d'objets (les instances) partageant des traits communs identifiables (Kassel, 1999). Il paraît donc incorrect d'un point de vue théorique d'assimiler un terme à une instance de concept. D'un point de vue pragmatique, une telle modélisation ne permet pas de représenter la polysémie éventuelle d'un terme puisqu'il ne peut être instance de deux concepts disjoints.

4 Description du modèle proposé

Les différents modèles existants sont donc insuffisants à représenter correctement la partie terminologique associée à une ontologie. Leurs limites nous ont permis de dégager quelques principes sur lesquels baser notre modèle. Tout d'abord, il est nécessaire de matérialiser la notion de terme de manière à pouvoir la manier aussi aisément qu'un concept. Ceci permettra notamment de pouvoir modéliser un nombre quelconque d'informations relatives à l'ancrage des concepts dans la langue. De plus, on a constaté qu'établir un lien de classe à instance entre les deux notions ne permet pas une reproduction fidèle de certains phénomènes linguistiques. Nous exposons ci-après une manière de stocker les termes en OWL-DL, sans qu'il y ait besoin d'étendre la syntaxe du langage ontologique. Nous verrons ensuite comment nous proposons de relier terme et concept dans le nouveau modèle.

4.1 La représentation du terme

Nous commençons par réifier le terme en le représentant sous forme de `owl:Class`. Le plus haut niveau d'abstraction de l'ontologie permet alors de faire la distinction entre un objet `Concept` et un objet `Terme`. De façon à respecter autant que possible la norme TMF présentée en 2.3.1, nous classons les termes selon leur langue d'origine. Enfin, comme OWL n'utilise pas l'hypothèse de nom unique, nous déclarons toutes les classes de l'arbre terminologique comme étant mutuellement disjointes. Ainsi qu'on

[4] Cet environnement de développement fournit à la fois un modèle pour systèmes de traitement de langage naturel, une interface de programmation (API) et un environnement graphique permettant d'effectuer des traitements linguistiques sur des corpus de textes.

peut le voir sur la figure 2, le modèle d'ontologie que nous proposons enrichit celui de OWL en séparant l'ontologie proprement dite (sous la classe OWL `Concept`) d'une hiérarchie terminologique de profondeur maximale 3 et dont les feuilles représentent les termes.

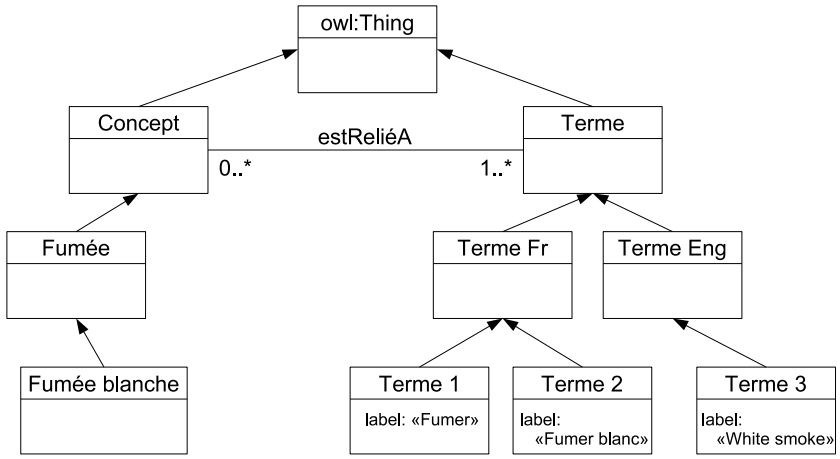

FIG. 2 – *Représentation simplifiée du modèle* [5]

Il est intéressant de noter que dans la norme ISO 12620 (sur laquelle s'appuie TMF), un terme abrégé est considéré comme distinct de sa forme complète. Afin de ne pas multiplier le nombre de termes à manipuler, nous préférons rassembler les variantes (abréviation, omission de mot) sous un même terme comme dans (Aussenac-Gilles, 1999). Dans le cadre de notre étude, nous avons choisi de ne représenter comme attribut du terme que sa localisation dans le corpus (un identifiant de texte et un identifiant de position). Toutefois, comme le terme est réifié dans notre modèle, il est aisé de lui rajouter de nouvelles propriétés (e.g. catégorie syntaxique, morphologie) sous forme de `owl:DatatypeProperty`.

4.2 La modélisation des liens terme-concept

Une fois la structure des termes établie dans OWL, analysons comment les relier aux concepts. En OWL, la structure de donnée adéquate pour relier deux classes est la propriété d'objet (`owl:ObjectProperty`). Comme une propriété de ce type est orientée (elle a une classe domaine et une classe codomaine), il nous faut envisager les deux cas possibles et voir lequel des deux est le plus fidèle à la conceptualisation souhaitée et/ou le plus facilement représentable en OWL. Pour la suite, on rappellera qu'en OWL, lorsqu'une propriété d'objet P a pour domaine un concept C, alors celui-ci en fera hériter tous ses fils. Il est de plus possible pour tout fils de C de restreindre

[5] On remarquera que le lien entre terme et concept n'est pas encore dirigé, cette discussion est abordée en 4.2.

le codomaine C' de P à un fils de C' (et tous les descendants du fils de C hériteront de cette restriction).

Considérons d'abord le cas d'une propriété *dénotéPar* ayant pour classe domaine un concept C et pour classe codomaine un terme. Si l'on prend un fils de C, celui-ci sera dénoté par au plus les mêmes termes que son père. D'un point de vue conceptuel, ce résultat n'est pas satisfaisant puisque l'on souhaite pouvoir associer des termes différents à deux concepts reliés hiérarchiquement. On pourrait imaginer d'associer aux termes dénotant C tous les termes dénotant ses fils mais un autre problème survient alors : on est incapable de connaître directement les termes dénotant C et uniquement lui (si ce n'est en parcourant l'ensemble de ses fils).

Plaçons nous maintenant dans la situation inverse où terme et concept sont reliés par une propriété *dénote* dirigée du terme vers le concept. On constate un premier avantage à cette modélisation : comme seules les feuilles de l'arbre terminologique sont des termes, il n'y aura pas de problème d'héritage de la propriété *dénote*. Du fait de la propriété d'héritage des classes en OWL, un tel choix entraîne qu'un terme dénotant un concept C dénote aussi les fils de C. Ce phénomène ne fait que refléter la réalité dans le sens où l'occurrence d'un terme dénotant un concept C_0 peut faire indirectement référence à une instance d'un concept C_1, fils de C_0. Par exemple, dans la phrase "*Le compteur de vitesse ne s'allume pas au démarrage*", on repère une occurrence du terme "*compteur de vitesse*" qui dénote le concept *Tachymètre*. Toutefois, on peut déduire grâce au contexte que l'instance de terme désigne une instance d'un concept plus précis, à savoir *Tachymètre digital*. Nous avons donc choisi de retenir cette solution, illustrée sur la figure 3.

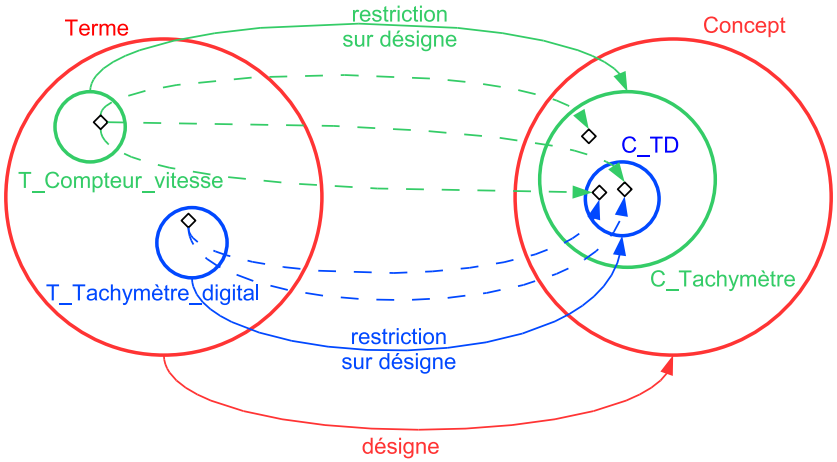

FIG. 3 – *Lien Terme-Concept*

5 Vers un éditeur de RTO

Comme nous étudions actuellement les bénéfices d'une mise en parallèle de l'enrichissement d'une ontologie à finalité annotative avec la phase d'indexation de textes, nous avons cherché un environnement capable à la fois d'éditer une ontologie avec une composante terminologique et de l'utiliser dans une tâche d'indexation semi-automatique. Nous avons arrêté notre choix sur Protégé-OWL pour sa légèreté d'exécution, son architecture ouverte (facilement extensible) et pour sa bonne ergonomie.

Comme son nom l'indique, Protégé-OWL est une extension de l'éditeur d'ontologie Protégé [6] pour manipuler le format OWL. Elle permet notamment de visualiser, d'éditer classes et propriétés OWL, de communiquer avec des raisonneurs logiques ou de peupler l'ontologie d'instances trouvées dans des documents. Visuellement, le logiciel se présente sous la forme de plusieurs onglets permettant l'accès à différents types d'information : on peut citer entre autres les onglets *OWLClasses*, *Properties* (avec une subdivision supplémentaire pour chaque type de propriété OWL) et *Individuals*.

Nous avons décidé de nous baser sur le plugin IAnnotate [7] qui joue le rôle d'intermédiaire entre les résultats de la phase d'indexation sémantique et l'ontologie : il permet de visualiser les textes indexés et d'éditer les annotations associées (figure 4). Grâce au modèle proposé, on peut assigner l'information d'indexation directement au terme et non plus au concept. De fait, comme IAnnotate modélise les annotations comme des propriétés d'individus, il nous suffit de le contraindre à n'associer les occurrences des termes qu'aux instances des classes de type `Terme`.

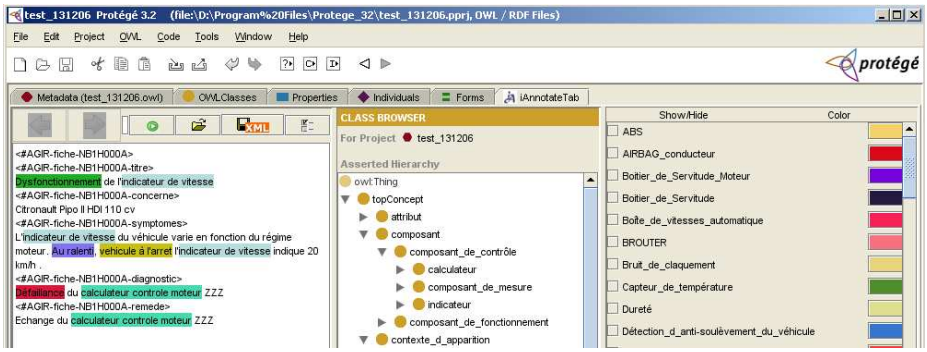

FIG. 4 – *Le plugin Iannotate dans l'environnement Protégé-OWL*

Nous souhaitons très prochainement réaliser une extension à Protégé-OWL. Celle-ci consistera en l'ajout de nouvelles fonctionnalités sur l'onglet `OWLClasses` (comme la possibilité de visualiser les termes associés à chaque concept) ainsi qu'en la création d'un nouvel onglet `Terms` permettant l'affichage spécifique des informations liées à la composante terminologique de l'ontologie. Un des avantages du modèle que nous avons

[6] `http://protege.stanford.edu/`

[7] `http://www.dbmi.columbia.edu/~cop7001/iAnnotateTab/iannotate.htm`

proposé réside dans la possibilité de manipuler la terminologie dans n'importe quel éditeur d'ontologie en OWL. Une interface dédiée serait néanmoins appropriée pour aider le modélisateur dans sa tâche. Elle devra permettre d'afficher la liste des termes dans une langue, ajouter ou supprimer un terme, et de modifier la valeur des attributs associés à chaque terme. On peut aussi imaginer qu'elle offre la possibilité d'importer des informations depuis des terminologies au format TMF. Enfin, comme nous le soulignons en 2.3.1, on peut choisir pour les termes de modéliser des propriétés de nature très différente selon l'objectif de modélisation recherché. Il serait donc bon que l'interface de présentation du terme s'adapte automatiquement en fonction des propriétés modélisées.

Outre son utilité pour l'étude de l'indexation sémantique, on constate que cette extension peut facilement devenir un environnement d'édition d'ontologie à composante terminologique. Notre modèle permet en effet de manipuler les informations lexicales et sémantiques nécessaires à la construction à partir de textes d'une RTO et IAnnotate s'avère un moyen simple d'accéder aux textes. Un tel éditeur pourrait constituer un support pour une partie de la méthode développée dans (Aussenac-Gilles *et al.*, 2003) (et mise en application dans Terminae), moyennant la disponibilité de logiciels de TAL ou l'accès à leurs résultats pour assurer l'analyse de textes préconisée. On pourrait alors évaluer et valider de façon rigoureuse le modèle que nous avons proposé dans cet article.

6 Conclusion et perspectives

Nous avons exposé une analyse de plusieurs modèles de représentation en OWL de la partie terminologique associée à une ontologie. Après avoir constaté leurs limites pour représenter certains phénomènes linguistiques comme l'anaphore, la polysémie ou le multilinguisme, nous avons proposé une solution qui présente une bonne expressivité terminologique tout en respectant la syntaxe classique du sous-langage OWL-DL. Ce modèle possède ainsi les avantages d'être manipulable dans n'importe quel éditeur important le format OWL et de ne pas entraver l'utilisation conjointe de raisonneurs. D'un point de vue pratique, une ontologie construite préalablement avec le logiciel Terminae a été correctement convertie dans ce modèle dans le but d'être manipulée dans Protégé-OWL. Dans un cadre d'indexation sémantique, nous travaillons actuellement sur la mise au point d'une extension à cet environnement qui permettra de manipuler facilement les termes, leurs propriétés, ainsi que le corpus à indexer. Cette extension pourrait par la suite trouver une utilité plus générale, comme par exemple permettre la construction et la maintenance de RTO à partir de textes. Une autre perspective consisterait à évaluer concrètement dans quelles proportions notre modèle respecte la norme TMF et travailler à une meilleure compatibilité entre les formats. Enfin, on pourrait s'interroger sur la capacité (et l'utilité) de ce modèle à représenter des structures plus complexes permettant l'accès aux concepts dans le texte comme les patrons d'extraction.

Références

AFNOR (1987). Vocabulaire de la documentation. In *Les Dossiers de la normalisation*. ISSN 0297-4827.

AUSSENAC-GILLES N. (1999). Gediterm : un logiciel pour gérer des Bases de Connaissances Terminologiques . *Terminologies nouvelles (revue internationale francophone)*, **19**, 111–123.

AUSSENAC-GILLES N., BIÉBOW B. & SZULMAN S. (2003). D'une méthode à un guide pratique de modélisation de connaissances à partir de textes. In F. ROUSSELOT, Ed., *Actes des 5e rencontres Terminologie et IA (TIA 2003)*, p. 41–53.

AUSSENAC-GILLES N. & CONDAMINES A. (2001). *Entre textes et ontologies formelles : les bases de connaissances terminologiques*, In *Ingénierie et capitalisation des connaissances*, p. 153–177. Hermes.

BAGET J., CANAUD E., EUZÉNAT J. & SAÏD-HACID M. (2004). Les langages du Web Sémantique. *Revue I3*, **Hors Série 2004**.

BERNERS-LEE T. (1999). *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Harper San Francisco.

BONTCHEVA K., TABLAN V., MAYNARD D. & CUNNINGHAM H. (2004). Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Engineering*, **10**(3/4), 349–373.

CHARLET J. (2002). L'Ingénierie des Connaissances : développements, résultats et perspectives pour la gestion des connaissances médicales. Mémoire d'habilitation à diriger des recherches en Informatique de l'université de Pierre et Marie Curie.

KASSEL G. (1999). PHYSICIAN is a role played by an object, whereas SIGN is a role played by a concept. In *Proc. of the IJCAI'99 Workshop on Ontologies and Problem-Solving Methods : Lessons Learned and Future Trends*, p. 61–69.

MAEDCHE A. (2002). *Ontology learning for the Semantic Web*. Kluwer Academic Publisher.

MEYER I., SKUCE D., BOWKER L. & ECK K. (1992). Towards a new generation of terminological resources : an experiment in building a terminological knowledge base. In *Proc. of 13th International Conference on Computational Linguistics*, p. 956–960.

MOTHE J. (2000). Recherche et exploration d'informations - découverte de connaissances pour l'accès à l'information. Mémoire d'habilitation à diriger des recherches en Informatique de l'université Paul Sabatier de Toulouse.

PRIÉ Y. (2000). Sur la piste de l'indexation conceptuelle de documents. une approche par l'annotation. *Document Numérique*, **4**(162), 11–35.

REYMONET A., AUSSENAC-GILLES N. & THOMAS J. (2006). Tâche, domaine et application : influences sur le processus de modélisation de connaissances. In *Actes des 17e journées francophones d'ingénierie des connaissances*.

ROMARY L. (2001). An abstract model for the representation of multilingual terminological data : TMF - Terminological Markup Framework. In *Proc. of TAMA*.

SZULMAN S. & BIÉBOW B. (2004). OWL et Terminae. In *Actes des 15es journées francophones d'ingénierie des connaissances*, p. 41–52 : Presses Universitaires de Grenoble.

SZULMAN S., BIÉBOW B. & AUSSENAC-GILLES N. (2002). Structuration de Terminologie à l'aide d'outils de TAL avec TERMINAE. In *Traitement Automatique des Langues*, volume 43, p. 103–128.

W3C (2004). Web Ontology Language OWL. http ://www.w3.org/2004/OWL/.